Lighting

- Goal
  - Model the interaction of light with surfaces to render realistic images
  - Generate per (pixel/vertex) color
Factors

- Light sources
  - Location, type & color
- Surface materials
  - How surfaces reflect light
- Transport of light
  - How light moves in a scene
- Viewer position

Illumination Models/Algorithms

- Local illumination - Fast
  - Ignore real physics, approximate the look
  - Interaction of each object with light
    - Compute on surface (light to viewer)
- Global illumination – Slow
  - Physically based
  - Interactions between objects
The big picture (basic)

- Light: energy in a range of wavelengths
  - White light – all wavelengths
  - Colored (e.g. red) – subset of wavelengths
- Surface “color” – reflected wavelength
  - White – reflects all lengths
  - Black – absorbs everything
  - Colored (e.g. red) absorbs all but the reflected color
- Multiple light sources add (energy sums)

Materials

- Surface reflectance:
  - Illuminate surface point with a ray of light from different directions
  - How much light is reflected in each direction?
Basic Types

Most surfaces exhibit complex reflectances that vary with incident and reflected directions. Model with combination known as **BRDF**:

**Reflectance Distribution Function**

- BRDF: Bidirectional Reflectance Distribution Function

**Reflectance Distribution Model**

- Most surfaces exhibit complex reflectances
  - Vary with incident and reflected directions.
  - Model with combination known as **BRDF**

**Reflectance Distribution Function**
Lambert’s “Law”

Intuitively: cross-sectional area of the “beam” intersecting an element of surface area is smaller for greater angles with the normal.
Computing Diffuse Reflection

- Depends on **angle of incidence**: angle between surface normal and incoming light
  - $I_{\text{diffuse}} = k_d \cdot I_{\text{light}} \cdot \cos(\theta)$
- In practice use vector arithmetic
  - $I_{\text{diffuse}} = k_d \cdot I_{\text{light}} \cdot (n \cdot l)$
- Always normalize vectors used in lighting
  - $n$, $l$ should be unit vectors
- Scalar (B/W intensity) or 3-tuple (color)
  - $k_d$: diffuse coefficient, surface color
  - $I_{\text{light}}$: incoming light intensity
  - $I_{\text{diffuse}}$: outgoing light intensity (for diffuse reflection)

Diffuse Lighting Examples

- Lambertian sphere from several lighting angles:

![Lambertian sphere from several lighting angles](image)

- need only consider angles from $0^\circ$ to $90^\circ$
Physics of Specular Reflection

- Geometry of specular (perfect mirror) reflection
  - Snell’s law

Snell’s law = perfect mirror-like surfaces
- But ..
  - few surfaces exhibit perfect specularity
  - Gaze and reflection directions never EXACTLY coincide
- Expect **most** reflected light to travel in direction predicted by Snell’s Law
- But some light may be reflected in a direction slightly off the ideal reflected ray
- As angle from ideal reflected ray increases, we expect less light to be reflected

Empirical Approximation
Empirical Approximation

- Angular falloff

\[
\begin{align*}
\vec{l} & \rightarrow \vec{n} \\
\theta_j & \\
\vec{r}
\end{align*}
\]

- How to model this falloff?

Phong Lighting

- Most common lighting model in computer graphics
  - (Phong Bui-Tuong, 1975)

\[
\begin{align*}
I_{\text{specular}} & = k_s I_{\text{light}} (\cos \phi)^{n_s} \\
I_{\text{specular}} & = k_s I_{\text{light}} (\vec{v} \cdot \vec{r})^{n_s}
\end{align*}
\]

\(\phi\): angle between \(\vec{r}\) and view direction \(\vec{v}\)

\(n_s\): purely empirical constant, varies rate of falloff

\(k_s\): specular coefficient, highlight color

no physical basis, "plastic" look
Phong Examples

varying light position

varying $n_s$

Blinn-Phong model (Jim Blinn, 1977)

Variation with better physical interpretation

$h$: halfway vector; $r$: roughness

$$I_{\text{specular}} = k_s \cdot (h \cdot n)^{1/r} \cdot I_{\text{light}}; \text{ with } h = (1 + v)/2$$
Light is **linear**
- If multiple rays illuminate the surface point the result is just the sum of the individual reflections for each ray

\[
\sum I_p (k_d (n \cdot l_p) + k_r (r_p \cdot v)^n)
\]

**Ambient Light**
- Non-directional light – environment light
- Object illuminated with same light everywhere
  - Looks like silhouette
- Illumination equation \( I = I_a k_a \)
  - \( I_a \) - ambient light intensity
  - \( k_a \) - fraction of this light reflected from surface
Light Source Types

- **Point Light**
  - light originates at a point

- **Directional Light (point light at infinity)**
  - light rays are parallel
  - Rays hit a planar surface at identical angles

- **Spot Light**
  - point light with limited angles

Light Source Falloff

- Quadratic falloff (point- and spot lights)
  - Brightness of objects depends on power per unit area that hits the object
  - The power per unit area for a point or spot light decreases quadratically with distance

\[
\text{Area } 4\pi r^2
\]

\[
\text{Area } 4\pi(2r)^2
\]
Illumination Equation

- For multiple light sources:

\[ I = I_a k_a + \sum p \frac{I_p}{A(d_p)} (k_d (n \cdot l_p) + k_l (r_p \cdot v)^\alpha) \]

- \( d_p \) - distance between surface and light source
- + distance between surface and viewer, \( A \) – attenuation function

\[ A(d) \propto \frac{1}{d^2} \]

Light

- Light has color
- Interacts with object color (r,g,b)

\[ I = I_a k_a \]
\[ I_a = (I_{ar}, I_{ag}, I_{ab}) \]
\[ k_a = (k_{ar}, k_{ag}, k_{ab}) \]
\[ I = (I_r, I_g, I_b) = (I_a k_{ar}, I_a k_{ag}, I_a k_{ab}) \]

- Blue light on white surface?
- Blue light on red surface?

\[ I_a = (I_{ab}, I_{ag}, I_{ar}) \]
\[ k_a = (I_r, I_g, I_b) \]
Light and Material Specification

- Light source: amount of RGB light emitted
  - value = percentage of full intensity, e.g., (1.0, 0.5, 0.5)
  - every light source emits ambient, diffuse, and specular light
- Materials: amount of RGB light reflected
  - value represents percentage reflected e.g., (0.0, 1.0, 0.5)
- Interaction: multiply components
  - Red light (1,0,0) x green surface (0,1,0) = black (0,0,0)

When to apply Lighting Model?

- per polygon
  - “flat shading”
- per vertex
  - “Gouraud shading”
- per pixel
  - “per pixel lighting”, “Phong shading”
Colored Wireframes

Ambient Lighting
Per-Polygon Shading

Per Vertex shading
Per Pixel Shading

Curved Surfaces with Per-pixel Shading
Complex Lighting and Shading

Texture Mapping
Displacement Mapping

Reflection Mapping
Global Illumination

Subsurface scattering